

## Marker ve Özel Hafızalar

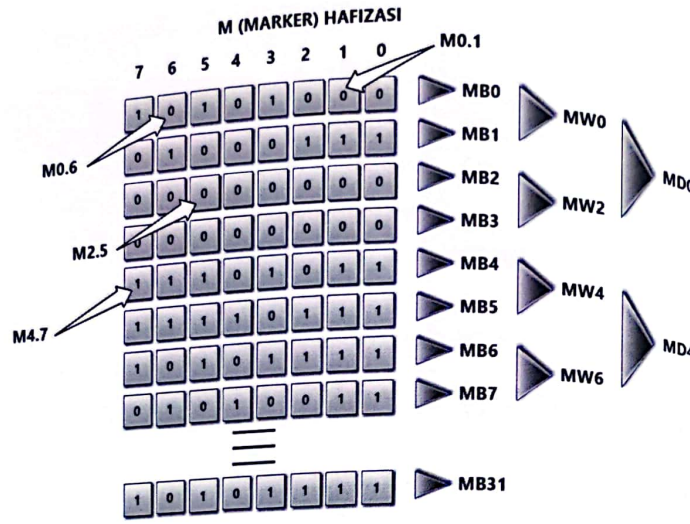
- Marker (Dahili Röle) Nedir?
- Marker'ler Nasıl Kullanılır?
- Özel Hafızalar Nedir?
- Özel Hafızalar Nasıl Kullanılır?
- Marker ve Özel Hafıza Uygulamaları

PLC programı yazarken işimizi kolaylaştıracak ve bilgisayar programlamadaki değişkenler gibi kullanılacak değişik türde hafızalar bulunmaktadır. Bu hafızalar **Marker**, **Special Memory** ve **Local Memory** gibi adlarla anılmaktadır. Bu bölümde **Marker** gibi özel hafızaları daha yakından tanıyıp daha pratik ve daha karmaşık programlar yapmayı öğreneceğiz.

### Marker (Yardımcı Röle)

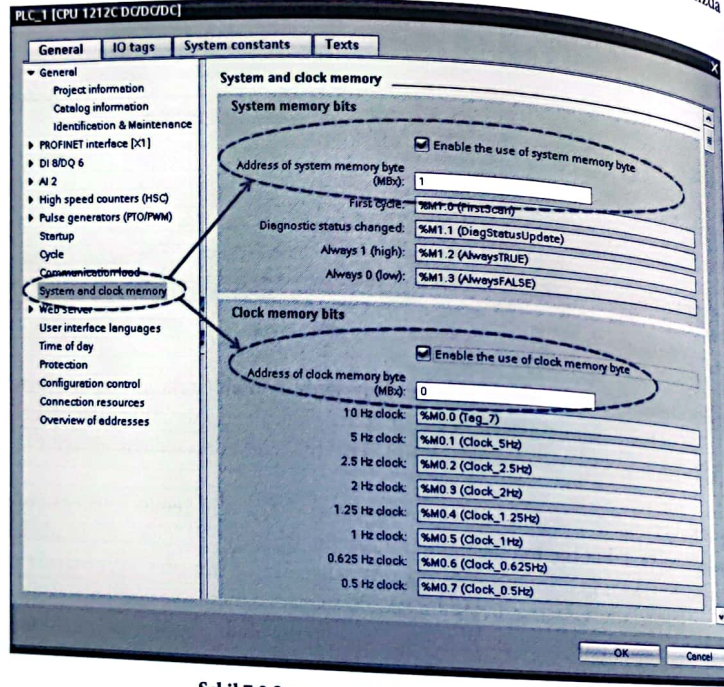
Marker ya da yardımcı röle olarak adlandırılan hafızalar belli değerleri geçici ya da kalıcı (retentive) olarak saklayan ve gerektiğinde kullanılan hafızalardır. Program içinde elde edilen bazı veriler ve işlem sonuçları marker olarak adlandırılan hafızalarda saklanır ve program içindeki diğer elemanlar tarafından kullanılır.

Siemens S7-1200 PLC'lerde, aşağıdaki şekilde görüldüğü gibi, modele göre değişen farklı boyutlarda marker hafızası bulunmaktadır.





System ve Clock Memory olarak adlandırılan hafızaların adresi kullanıcı tarafından belirlenir. Bu hafızaların adresinin belirlenmesi için önce **Project tree**→ **Sağ tuş**→ **Properties**→ **General**→ **System and clock memory** menüsüne gidilir. Daha sonra da **Enable** kutuları işaretlenir. **Enable** kutularının işaretlenmesiyle beraber bu adresler aktif hale gelir. Varsayılan olarak **System memory** MB1 adresini kullanırken, **Clock memory** MB0 adresini kullanır. İsterseniz **Enable** kutularının altında görülen kurullardan bu adreslerin değerini değiştirebilirsiniz. Biz bu kitabımızda varsayılan değerleri kullanacağız.



Şekil 7.3 System ve Clock memory ayarları

Yukarıdaki ayarların yapılmasıyla beraber M hafızasının 0. ve 1. baytları System ve Clock için ayrılmış oldu. Bunun sonucu olarak bu hafızalar program içinde veri saklama amacıyla kullanılmamalıdır. Aksi takdirde programın yanlış çalışmasına neden olursunuz.

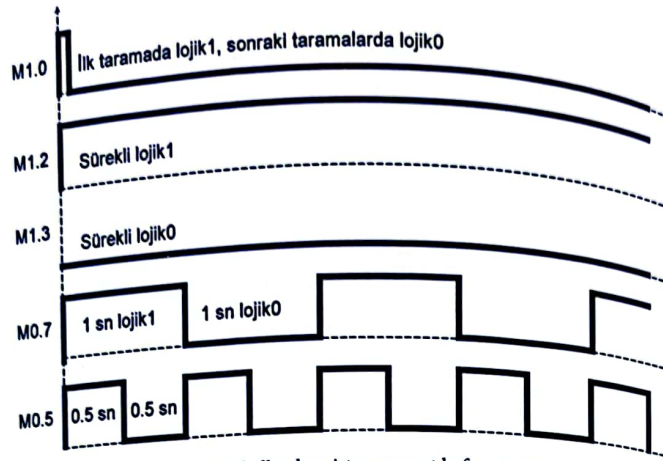
System hafızasındaki bitlerin açıklamalarını aşağıdaki tabloda görebilirsiniz. Bu tabloda System hafızasının MB1 olduğunu varsayıyoruz.

MB1	Açıklama
M1.0 (First Scan)	Bu bit ilk taramada lojik1'dir Daha sonra lojik0 olur. Başlangıç ayarlarının yapılması veya PLC'nin açılışıyla beraber bazı aygıtların çalışmasını sağlamak için kullanılır.
M1.2 (AlwaysTRUE)	Bu bit her zaman lojik1'dir. Kontakların lojik1 yapılması için kullanılabilir.
M1.3 (AlwaysFALSE)	Bu bit her zaman lojik0'dır. Kontakların lojik0 yapılması için kullanılabilir.

Clock hafızasındaki bitlerin açıklamalarını aşağıdaki tabloda görebilirsiniz. Bu tabloda Clock hafızasının MB0 olduğunu varsayıyoruz.

MB0	Açıklama
M0.0 (Clock_10Hz)	10 Hz frekansında kare dalga üretir. Yani, bu adresin değeri 1 saniyede 10 defa lojik0/lojik1 olarak durum değiştirir.
M0.1 (Clock_5Hz)	5 Hz frekansında kare dalga üretir.
M0.2 (Clock_2.5Hz)	2.5 Hz frekansında kare dalga üretir.
M0.3 (Clock_2Hz)	2 Hz frekansında kare dalga üretir.
M0.4 (Clock_1.25Hz)	1.25 Hz frekansında kare dalga üretir.
M0.5 (Clock_1Hz)	1 Hz frekansında kare dalga üretir. Yani, bu adresin değeri 1 saniyede 1 defa lojik0/lojik1 olarak durum değiştirir.
M0.6 (Clock_0.625Hz)	0.625 Hz frekansında kare dalga üretir.
M0.7 (Clock_0.5Hz)	0.5 Hz frekansında kare dalga üretir. Yani, bu adresin değeri 2 saniyede 1 defa lojik0/lojik1 olarak durum değiştirir.

Aşağıdaki şekilde sıklıkla kullanılan M1.0, M1.2, M1.3, M0.7 ve M0.5 adreslerinin sinyal grafiğini görebilirsiniz.



Şekil 7.4 Çok kullanılan sistem ve saat hafıza grafikleri

System ve Clock Memory hafıza alanlarının PLC'nin donanım alanına/konusuna girdiğini ve bu hafıza alanlarının kullanılabilmesi için Enable edildikten sonra Hardware olarak Download edilmesi gerektiğini, yani, PLC'de yapılan bütün donanım ayarlarının PLC'ye yüklenmesi gerektiğini unutmayın!!!

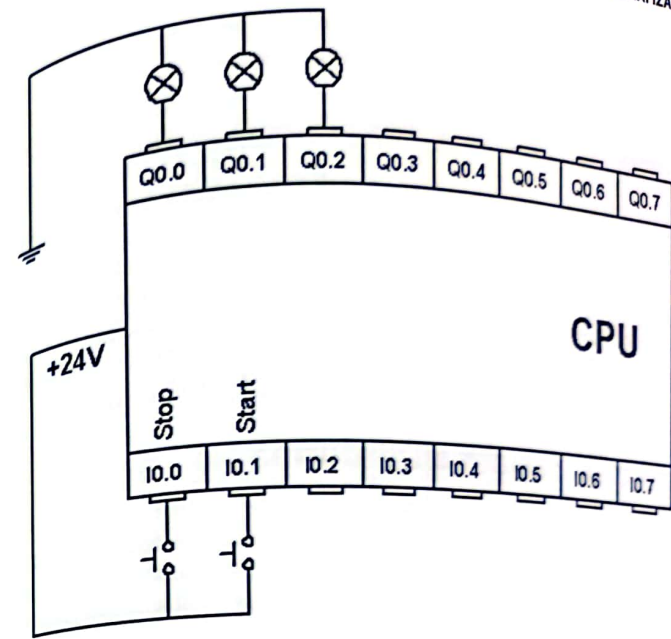
Aşağıdaki uygulamalarda System ve Clock hafızaları ile ilgili örnekler yapılarak konuların pekiştirilmesi amaçlanmıştır. Lütfen bu uygulamaları iyice inceledikten sonra uygulamaları test ederek konunun iyice anlaşılmasını sağlayınız.

## Uygulamalar

Bu bölümde öğrendiğimiz konuları pekiştirmek ve özel hafızaları daha iyi anlayabilmek için birkaç uygulama yapacağız. Lütfen tüm uygulamaları PLC'de ya da simülatörde deneyerek komutların çalışma mantığını iyi kavrayınız.

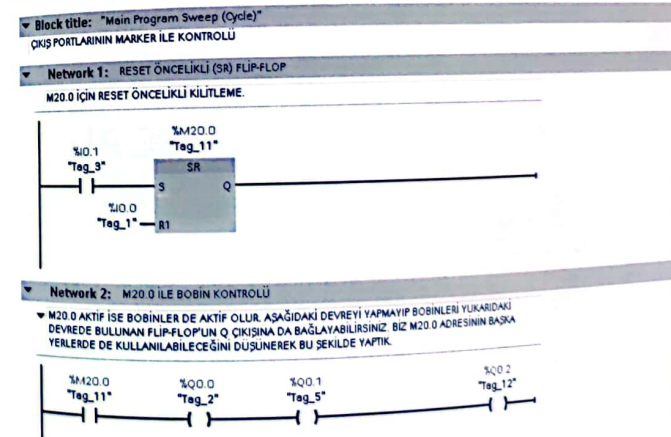
## Marker ile Sistem Kontrolü

Bu uygulamamızda bir marker kullanarak tüm sistemimizi kontrol edeceğiz. Sistemimizde 3 adet lamba veya makine olduğunu varsayacağız. Bu lambaların/makine-lerin çalışması veya durması sistemin tamamını kontrol eden Start/Stop butonları ile mümkün olacaktır. Uygulamanın PLC devre şeması aşağıdaki gibidir.



Şekil 7.5 PLC devre şeması

Q0.0, Q0.1 ve Q0.2 adresli lambaları I0.0 (Stop) ve I0.1 (Start) butonları ile kontrol edeceğiz. **Start** ve **Stop** butonları ile M20.0 adresini SET veya RESET edeceğiz. Q0.0, Q0.1 ve Q0.2 adresli lambalar M20.0 adresindeki duruma göre aktif veya pasif olacak. Buna göre yazmamız gereken program aşağıdaki gibidir.



Şekil 7.6 PLC programı

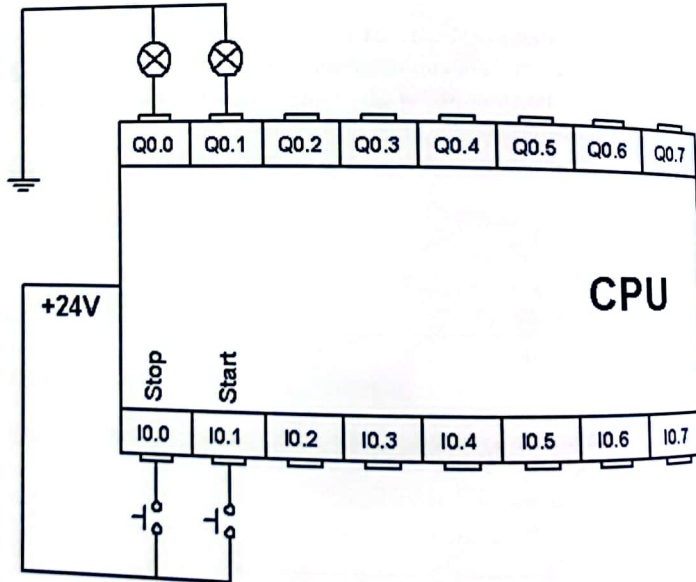


Yukarıdaki programı yazdıktan sonra **Download** komutunu kullanarak PLC'ye yükleyin. Daha sonra da **Monitor on/off** komutu ile izleme modunu aktif hale getirin. Son olarak, PLCSIM yazılımına I0.0 ve I0.1 kontakları ile Q0.0, Q0.1 ve Q0.2 bobinlerini ekleyerek Q0.0, Q0.1 ve Q0.2 adresli bobinlerin davranışını izleyin.

Programın ilk aşında (Network1) Reset öncelikli flip-flop kullanılmıştır. Eğer **Start** butonuna basılırsa I0.1 kontağı kapanır ve M20.0 adresini lojik1 yapar. M20.0 adresi lojik1 olunca Q0.0, Q0.1 ve Q0.2 bobinlerini kontrol eden kontaklar kapanır ve bobinler enerjilenir. Eğer **Stop** butonuna basılırsa I0.0 kontağı kapanır ve flip-flopun Reset girişine sinyal uygulanır. Bu sinyali alan flip-flop çıkışını, yani M0.0 adresini lojik0 yapar. M0.0 adresi lojik0 olunca Q0.0, Q0.1 ve Q0.2 bobinlerini kontrol eden kontaklar açılır ve bobinlerin lojik0 konumuna geçmesini sağlar.

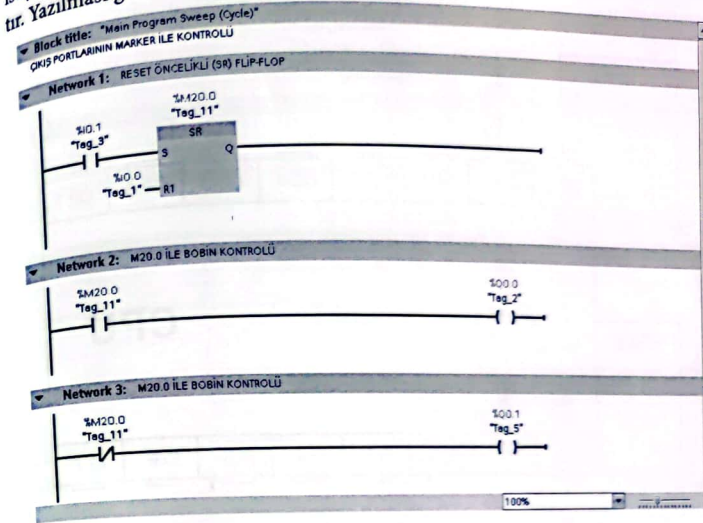
## Marker ile Başlatma/Durdurma

Yardımcı röleleri açık veya kapalı kontak kullanarak bazı aygıtları çalıştırabilir, bazı aygıtları ise durdurabilirsiniz. Bu uygulamamızda sistemi **Start** ettiğimizde Q0.0 aktif olurken Q0.1 pasif olacak. **Stop** ettiğimizde ise Q0.0 pasif olurken Q0.1 aktif olacak. Yani, PLC çıkışına bağlı cihazın biri çalışırken diğeri duracak. Uygulamanın devre şeması aşağıdaki gibidir.



Şekil 7.7 PLC devre şeması

**Start** butonuna basıldığında Q0.0 adresinin aktif, Q0.1 adresinin ise pasif olmasını istiyoruz. **Stop** butonuna basıldığında ise Q0.0 adresi pasif, Q0.1 adresi aktif olacaktır. Yazılması gereken program aşağıdaki gibidir.



Şekil 7.8 PLC programı

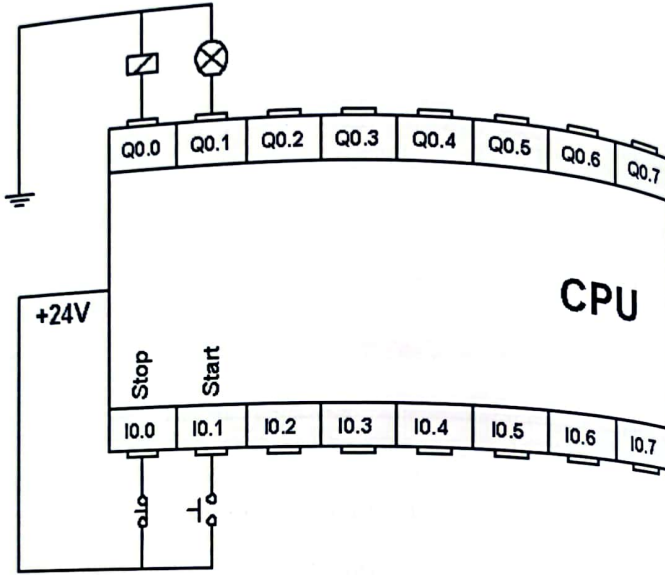
Yukarıdaki programı yazdıktan sonra **Download** komutunu kullanarak PLC'ye yükleyin. Daha sonra da **Monitor on/off** komutu ile izleme modunu aktif hale getirin. Son olarak, PLCSIM yazılımına I0.0 ve I0.1 kontakları ile Q0.0 ve Q0.1 bobinlerini ekleyerek Q0.0 ve Q0.1 adresli bobinlerin davranışını izleyin.

PLC çalıştığında M20.0 adresinin değeri lojik0'dır. M20.0 adresi lojik0 iken Q0.0 adresini kontrol eden kontak açık, Q0.1 adresini kontrol eden kontak kapalı, yani normal konumlarında olur. Bu durumda Q0.0 pasif, Q0.1 ise aktif olur. **Start** butonuna basılırsa M20.0 lojik1 durumuna geçer. M0.0 lojik1 olunca kontaklar konum değiştirir; Network2'deki kontak kapanır ve Network3'teki kontak açılır. Böylece Q0.0 aktif, Q0.1 pasif olur. **Stop** butonuna basıldığında ise başlangıç durumlarına dönerler.

## Makine ve Flaşör Uygulaması

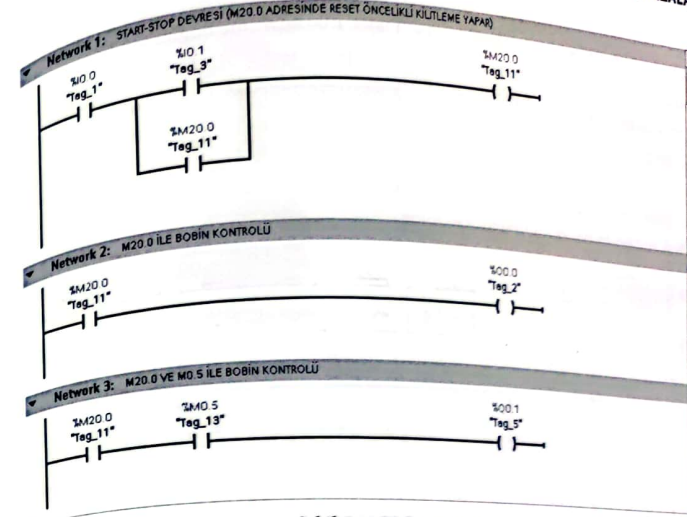
Bu uygulamada **Start** butonuna bastığımızda Q0.0 adresine bağlı bir röleyi çektireceğiz. Rölenin çekmesi ile beraber röle ile kumanda edilen bir makineyi de çalıştıracacağız. Makine çalışmaya başladığında röle bırakacak ve makine duracaktır. ya başlayacak. **Stop** butonuna bastığımızda röle sönecektir. Flaşörün yanıp sönmeye frekansı 1Hz olacaktır. Makinenin durmasıyla flaşör de sönecektir.

caktır. PLC'nin devre şeması aşağıdaki gibidir. I0.0 adresine bağlı olan butonun normalde kapalı (NC) olduğuna dikkat ediniz. Bu durum PLC çalıştığı anda I0.0 girişinin aktif (lojik1) olmasını sağlar. Dolayısıyla, simülasyonu da bu doğrultuda yapmanız gerekir.



Şekil 7.9 PLC devre şeması

Flaşördeki 1 Hz yanıp sönme frekansını elde etmek için M0.5 özel hafızasını kullanacağız. Bu hafıza 0.5 saniye lojik1, 0.5 saniye lojik0 olur. Lambanın önüne koyacağımız bir kontak ile lambanın 1 Hz yanıp sönmesini sağlayacağız. M0.5 hafızasının işlevini yerine getirebilmesi için **Project tree**→**PLC**→**Sağ tuş**→**Properties**→**General**→**System and clock memory** menüsündeki **Enable** kutucuklarını işaretlemeyi unutmayınız. PLC programı aşağıdaki gibidir.



Şekil 7.10 PLC programı

Yukarıdaki programı yazdıktan sonra **Download** komutunu kullanarak PLC'ye yükleyin. Daha sonra da **Monitor on/off** komutu ile izleme modunu aktif hale getirin. Son olarak, PLCSIM yazılımına I0.0, I0.1 ve M20.0 kontakları ile Q0.0 ve Q0.1 bobinlerini ekleyerek Q0.0 ve Q0.1 adresli bobinlerin davranışını izleyin.

I0.0 girişine bağlı olan **Stop** butonunun normalde kapalı kontak olduğunu unutmayınız. Bu nedenle PLC'yi çalıştırmadan önce simülatörde I0.0 adresini işaretleyerek lojik1 yapınız. Bu butona basılı olan durumu simüle etmek için I0.0 girişini lojik0 yapınız.

PLC'yi **RUN** konumuna aldıktan sonra **Start** butonuna bastığınızda Q0.0 aktif olacak, bununla birlikte Q0.1 adresine bağlı lamba 1 Hz'de yanıp sönecektir.

## Otomatik Makine Çalıştırma Uygulaması

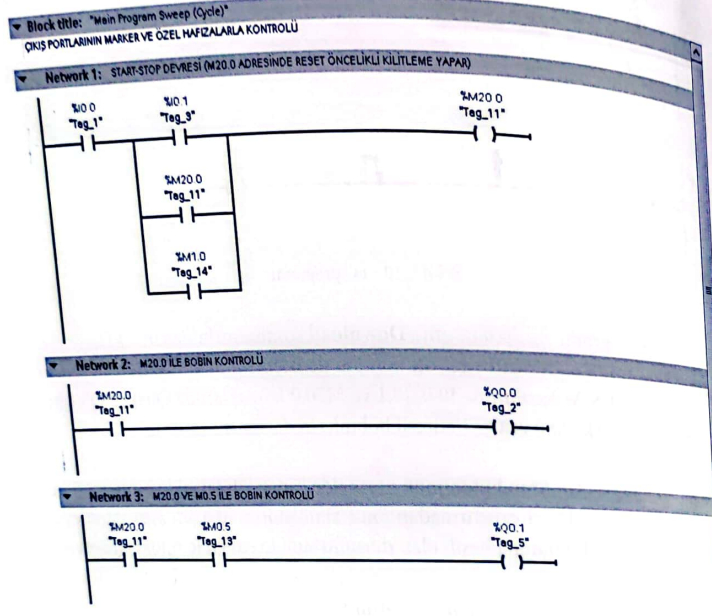
Bu uygulamada PLC'nin **RUN** konumuna alınmasıyla Q0.0 adresine bağlı rölenin otomatik olarak (herhangi bir butona basmadan) çekmesini sağlayacağız. Rölenin otomatik olarak çekilmesiyle beraber röle tarafından kontrol edilen cihaz çalışmaya başlayacaktır. Cihazın çalışmasıyla beraber bir flaşörün de çalışmasını sağlayacağız. **Stop** butonuna basıldığında ise cihaz duracaktır. Eğer isterseniz **Start** butonuna basarak röleyi tekrar çektirebileceğiz.

Hatırlanacağı gibi, **System** ve **Clock** hafızaların etkinleştirilmesi ve adresinin belirlenmesi için önce **Project tree**→**PLC**→**Sağ tuş**→**Properties**→**System**



and clock memory menüsüne gidilir. Daha sonra da **Enable** kutuları işaretlenir. Enable kutularının işaretlenmesiyle beraber bu adresler aktif hale gelir. Varsayılan olarak **System memory MB1** adresini kullanırken, **Clock memory MB0** adresini kullanır.

Bu uygulama için PLC devre şeması bir önceki uygulamayla aynıdır. Yazmamız gereken program ise aşağıdaki gibi olabilir.



Şekil 7.11 PLC programı

Yukarıdaki programı yazdıktan sonra **Download** komutunu kullanarak PLC'ye yükleyin. Daha sonra da **Monitor on/off** komutu ile izleme modunu aktif hale getirin. Son olarak, PLCSIM yazılımına I0.0, I0.1 ve M20.0 kontakları ile Q0.0 ve Q0.1 bobinlerini ekleyerek Q0.0 ve Q0.1 adresli bobinlerin davranışını izleyin.

Bir önceki uygulamadan farklı olarak **Start** butonuna paralel M1.0 adresli bir kontak eklenmiştir. Bu kontak PLC'nin ilk çevriminde lojik1, sonraki çevrimlerde ise lojik0 olur. M1.0 kontağı PLC'nin ilk çevriminde lojik1 olunca M20.0 aktif olur ve rölenin çekmesini sağlar.

# 8

## Kenar Tetikleme

- Kenar Tetikleme Nedir?
- Pozitif Kenar Tetikleme
- Negatif Kenar Tetikleme
- Kenar Tetikleme Uygulamaları

Genelde belli bir işlemin başlatılması herhangi bir butona bastığımızda gerçekleşir. Ama bazen bazı işlemlerin butona bastığımızda değil de butondan elimizi çektiğimizde gerçekleşmesini isteriz. Kenar tetikleme işlemleri genellikle sürekli gelen sinyallerin istenmediği yerlerde tercih edilir. Örneğin Reset öncelikli bir Flip-Flop ya da Set-Reset rölesi ile çalıştırılan bir motorun Reset girişine sürekli ON sinyali geldiğinde bu motoru Start etmek imkânsızdır. Bazen bir oparandın, bazen de bir mantıksal işlemin değerinin kısa süreli olması istenir. İşte bu ve bunun gibi durumlarda kenar tetiklemeleri kullanılır.

Bu bölümde önce Pozitif ve Negatif kenar tetiklemenin ne olduğunu öğreneceğiz. Daha sonra da konuyu pekiştirmek için uygulamalar yapacağız.

### Pozitif ve Negatif Kenar Nedir?

Aşağıdaki PLC şemasında görüldüğü gibi, I0.0 ve I0.1 adreslerine bağlı birer anahtar olduğunu düşünelim. Hatırlayacağınız gibi, anahtarlar butondan farklı olarak bırakıldıkları konumda kalırlar. Butonlar ise elinizi çektiğiniz anda eski konumlarına geri dönerler.